Database Design Patterns

Ovidiu Vasutiu and Florina Vasutiu

The purpose of this communication is to analyze the classical database design solutions to some classical problems like multi-language support or "history data" maintenance in software applications. We shall focus on the web-application and on the ERP Systems.

As internationalization is no longer a paper concepts, the applications address users that speak different languages - so multi-language support is one of the issues to be considered when designing an application. We have analyzed the way in which several applications implement the multi-language support.

Among of aspects to be considered when implementing the language database design are: the possibility to switch the language during the runtime, the measure in which the application data needs to be translated, the most important, the dynamism of the data and the data type.

If the translation is to be stored in the same table with the "original" data, or in "dictionary" table is to be discussed. If the data used in application is relatively static, than a single dictionary table (more dictionary tables grouped by different criteria) properly indexed is doing the job. For dynamic, different in type application data, different solutions involving auto-referred tables or more dictionary tables are used.

Maintaining the old version of the data, saving intermediary data and the archive of date to be rarely used are some aspects to be considered when we speak about "history data" maintenance. Depending about the amount of legacy data that the business needs required, different design are used and they are to be presented and subject to discussions, as a definitive pattern is not yet defined. The need of additional columns to the table or maybe the need of a parallel table might be noticed, but also auto-referred tables might be used.

The way in which the classical database operations: insert, update, delete are performed for each of this solutions, the optimal indexations of the tables and the advantages of each method are to be discussed. An exemplification of the applications of these remarks to the classical ERP solutions (SAP mostly) are to be presented and the recommendations for implementing web-applications are further to be discussed.

The modeling method used is, generally, the entity-relationship model, but the object-role modeling is going to be used were further details are needed.